PART OF THE SUPERVISORY CONTROLLER FOR THE OISTERWIJKSEBAAN-BRIDGE SPLC

FERDIE F.H. REIJNEN, TOBY R. ERENS, JOANNA M. VAN DE MORTEL-FRONCZAK, AND JACOBUS E. ROODA

In this document, an example is provided regarding the splitting of a supervisor for implementation on a safety PLC. For this, part of the supervisor for the Oisterwijksebaan bridge, located in Tilburg, The Netherland, is described. The plant consists of 14 components, modeled with finite automata (FAs) and Boolean input variables (BIVs) and there are 28 event-condition requirements. In Section 1 the case is described. Subsequently, in Section 2, the splitting is performed. In the appendix, a list of symbols and the implementation code is provided.

1. The Oisterwijksebaan bridge case

Figure. 1 shows a sketch of the Oisterwijksebaan-bridge. This is a rotating bridge that can be opened whenever vessels want to pass. The bridge consists of the following components: two stop signs SS1 and SS2, two boom barriers BB1 and BB2, two boom barrier lights BL1 and BL2, and a bridge deck BD. Additionally, a graphical user interface is present, allowing an operator to send commands to the bridge. In the remainder of this section, the component models are provided

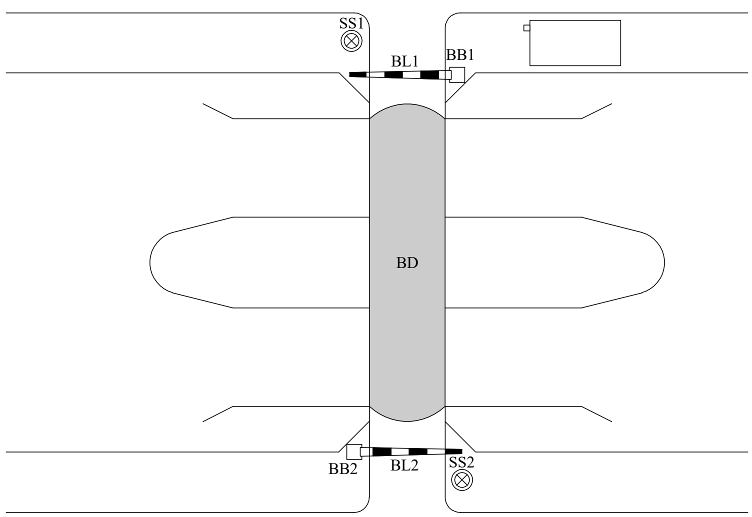


Figure 1. Oisterwijksebaan-bridge.

Figure 2 shows the model of the actuator that enables both stop signs simultaneously. Both stop signs have a sensor that measures whether the lamp is activated. These sensors are represented by two BIVs: `S_SS1_On` and `S_SS2_On`.

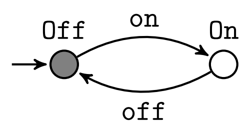


FIGURE 2. Stop sign actuator P_{SS}.

Each boom barrier consists of a bidirectional actuator, shown in Figure 3. Each boom barrier has two sensors that measure whether the barrier is fully closed or fully open. These sensors are represented by BIVs: `S_BB1_Open`, `S_BB1_Closed`, `S_BB2_Open`, and `S_BB2_Open`.

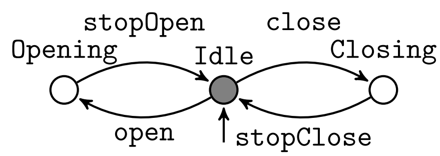


FIGURE 3. Boom barrier actuator $P_{\mathrm{BB}X}, X \in \{1, 2\}$.

Each boom barrier contains a light BL. One actuator enables both lights simultaneously, shown in Figure 4. Each light has a sensor that measures whether the lamp is activated. These sensors are represent by two BIVs: `S_BL1_On` and `S_BL2_On`.

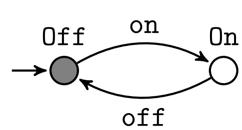


FIGURE 4. Barrier light actuator P_{BL}.

The bridge deck consists of a bidirectional actuator, shown in Figure 5. The bridge deck has two sensors that measure when it is fully closed or fully open. These sensors are represented by two BIVs: `S_BD_Open` and `S_BD_Closed`.

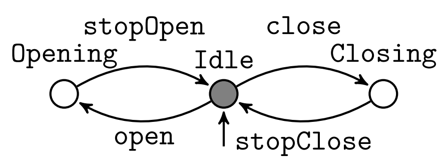


FIGURE 5. Bridge deck actuator P_{BD}.

Lastly, there is a GUI present, shown in Figure 6. A command (via a mouse click) is modeled as an uncontrollable event. Whenever a command activates an action, a controllable event represents that action being completed. The possible commands are: stopping land traffic, releasing land traffic, closing the barriers, opening the barriers, opening the bridge, and closing the bridge.

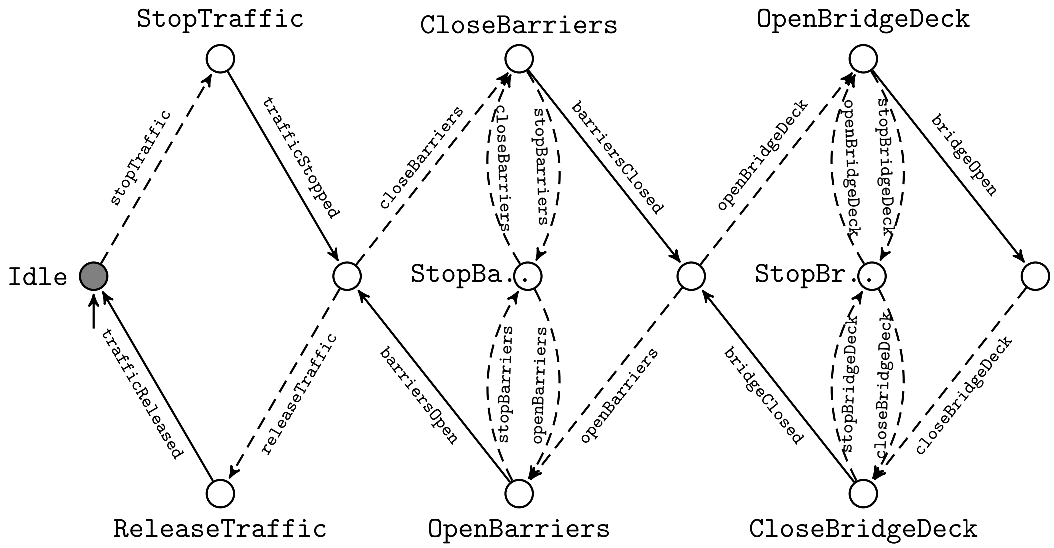


FIGURE 6. GUI P_{GUI}.

The components are connected to the variables in the output image of the PLC, via a hardware mapping. The hardware mapping is shown in Table 1. For each output image variable, one event is defined for setting the value to **T** and one event for setting the value to **F**.

TABLE 1. Hardware mapping of the actuators T_Q.

Output image variable	Event for **T**	Event for **F**
q_{stopsign}	P_{SS}.on	P_{SS}.off
q_{bb1Open}	P_{BB1}.open	P_{BB1}.stopOpen
q_{bb1Close}	P_{BB1}.close	P_{BB1}.stopClose
q_{bb2Open}	P_{BB2}.open	P_{BB2}.stopOpen
q_{bb2Close}	P_{BB2}.close	P_{BB2}.stopClose
$q_{\text{barrierlight}}$	P_{BL}.on	P_{BL}.off
$q_{\text{bridgeOpen}}$	P_{BD}.open	P_{BD}.stopOpen
$q_{\text{bridgeClose}}$	P_{BD}.close	P_{BD}.stopClose

Tables 2 and 3 show the regular requirements $\mathcal{R}'_{\text{R}}$ and the safety requirements $\mathcal{R}'_{\text{S}}$, respectively. The portioning is based on as risk assessment study, conducted by safety experts.

Table 2. Regular requirements $\mathcal{R}'_{\mathrm{R}}$.

	Event name	Condition
R1	P_{GUI}.trafficStopped	S_SS1_On $\wedge$ S_SS2_On
R2	P_{GUI}.trafficReleased	$\neg$S_SS1_On $\wedge$ $\neg$S_SS2_On
R3	P_{GUI}.barriersClosed	S_BB1_Closed $\wedge$ S_BB2_Closed
R4	P_{GUI}.barriersOpen	S_BB1_Open $\wedge$ S_BB2_Open
R5	P_{GUI}.bridgeOpen	S_BD_Open
R6	P_{GUI}.bridgeClosed	S_BD_Closed
R7	P_{SS}.on	P_{GUI}.StopTraffic
R8	P_{SS}.off	P_{GUI}.ReleaseTraffic
R9	P_{BB1}.open	P_{GUI}.OpenBarriers
R10	P_{BB1}.close	P_{GUI}.CloseBarriers
R11	P_{BB1}.stopOpen	P_{GUI}.StopBarriers $\vee$ S_BB1_Open
R12	P_{BB1}.stopClose	P_{GUI}.StopBarriers $\vee$ S_BB1_Closed
R13	P_{BB2}.open	P_{GUI}.OpenBarriers
R14	P_{BB2}.close	P_{GUI}.CloseBarriers
R15	P_{BB2}.stopOpen	P_{GUI}.StopBarriers $\vee$ S_BB2_Open
R16	P_{BB2}.stopClose	P_{GUI}.StopBarriers $\vee$ S_BB2_Closed
R17	P_{BL}.on	P_{GUI}.StopTraffic
R18	P_{BL}.off	P_{GUI}.ReleaseTraffic
R19	P_{BD}.open	P_{GUI}.OpenBridgeDeck
R20	P_{BD}.close	P_{GUI}.CloseBridgeDeck
R21	P_{BD}.stopOpen	P_{GUI}.StopBridgeDeck $\vee$ S_BD_Open
R22	P_{BD}.stopClose	P_{GUI}.StopBridgeDeck $\vee$ S_BD_Closed

Table 3. Safety requirements $\mathcal{R}'_{\mathrm{S}}$.

	Event name	Condition
R23	P_{SS}.off	S_BB1_Open $\wedge$ S_BB2_Open
R24	P_{BB1}.open	S_BD_Closed
R25	P_{BB1}.close	S_SS1_On $\wedge$ S_SS2_On
R26	P_{BB2}.open	S_BD_Closed
R27	P_{BB2}.close	S_SS1_On $\wedge$ S_SS2_On
R28	P_{BD}.open	S_BB1_Closed $\wedge$ S_BB2_Closed

When performing supervisor synthesis, it has shown that the plant in combination with the requirements is safe, nonblocking, controllable, and maximally permissive.

2. Splitting the supervisor

In this section, the method as described in Section 4.2 of the paper is followed. The steps are as follows.

(a) It is verified that the plant is a product system and that the condition given in Eq. 3 holds, i.e., all safety requirements depend on BIVs.

(b) Sets I_{S} and I_{R} are derived (Eqs. 4 and 5):

$$I_{\mathrm{S}} = \{\texttt{S\_SS1\_On}, \texttt{S\_SS2\_On}, \texttt{S\_BB1\_Open}, \texttt{S\_BB1\_Closed}, \texttt{S\_BB2\_Open}, \texttt{S\_BB2\_Closed}, \texttt{S\_BD\_Closed}\}$$
$$I_{\mathrm{R}} = \{\texttt{S\_BL1\_On}, \texttt{S\_BL2\_On}, \texttt{S\_BD\_Open}\}$$

(c) Sets Q_{S} and Q_{R} are derived (Eqs. 6 and 7).

$$Q_{\mathrm{S}} = \{q_{\mathrm{stopsign}}, q_{\mathrm{bb1Open}}, q_{\mathrm{bb1Close}}, q_{\mathrm{bb2Open}}, q_{\mathrm{bb2Close}}, q_{\mathrm{bridgeOpen}}\}$$
$$Q_{\mathrm{R}} = \{q_{\mathrm{barrierlight}}, q_{\mathrm{bridgeClose}}\}$$

(d) Sets $\mathcal{P}_{\mathrm{S}}$ and $\mathcal{P}_{\mathrm{R}}$ are derived (Eqs. 8 and 9).

$$\mathcal{P}_{\mathrm{S}} = \{P_{\mathrm{SS}}, P_{\mathrm{BB1}}, P_{\mathrm{BB2}}, P_{\mathrm{BD}}\}$$
$$\mathcal{P}_{\mathrm{R}} = \{P_{\mathrm{GUI}}, P_{\mathrm{BL}}\}$$

(e) Sets Σ_{S} and Σ_{R} are derived (Eqs. 10 and 11). These events are simply the events belonging to the safety and regular components above.

(f) Regular requirement set $\mathcal{R}'_{\mathrm{R}}$ is split (Eqs. 12 and 13).

$$\mathcal{R}^{\mathrm{S}}_{\mathrm{R}} = \{\text{R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R19, R20, R21, R22}\}$$
$$\mathcal{R}^{\mathrm{R}}_{\mathrm{R}} = \{\text{R1, R2, R3, R4, R5, R6, R17, R18}\}$$

(g) Regular requirement set for the safety events $\mathcal{R}^{\mathrm{S}}_{\mathrm{R}}$ is split (Eqs. 14 and 15).

$$\mathcal{R}^{\mathrm{SS}}_{\mathrm{R}} = \{\text{R11, R12, R15, R16, R22}\}$$
$$\mathcal{R}^{\mathrm{SR}}_{\mathrm{R}} = \{\text{R7, R8, R9, R10, R13, R14, R19, R20, R21}\}$$

(h) Requirements in $\mathcal{R}^{\mathrm{SR}}_{\mathrm{R}}$ are merged (Eq. 16). The result is given in Table 4.

TABLE 4. R2S communication.

Variable name	Condition
$c_{\texttt{SS.on}}$	P_{GUI}.`StopTraffic`
$c_{\texttt{SS.off}}$	P_{GUI}.`ReleaseTraffic`
$c_{\texttt{BB1.open}}$	P_{GUI}.`OpenBarriers`
$c_{\texttt{BB1.close}}$	P_{GUI}.`CloseBarriers`
$c_{\texttt{BB2.open}}$	P_{GUI}.`OpenBarriers`
$c_{\texttt{BB2.close}}$	P_{GUI}.`CloseBarriers`
$c_{\texttt{BD.open}}$	P_{GUI}.`OpenBridgeDeck`
$c_{\texttt{BD.close}}$	P_{GUI}.`CloseBridgeDeck`
$c_{\texttt{BD.stopOpen}}$	P_{GUI}.`StopBridgeDeck` $\vee$ `S_BD_Open`

(i) Requirement set $\mathcal{R}_D$ is added, Table 5 (Eq. 17).

TABLE 5. $\mathcal{R}_D$ requirements.

	Event name	Condition
R29	$P_{\mathrm{SS}}.\mathtt{on}$	$v_{\mathtt{SS.on}}$
R30	$P_{\mathrm{SS}}.\mathtt{off}$	$v_{\mathtt{SS.off}}$
R31	$P_{\mathrm{BB1}}.\mathtt{open}$	$v_{\mathtt{BB1.open}}$
R32	$P_{\mathrm{BB1}}.\mathtt{close}$	$v_{\mathtt{BB1.close}}$
R33	$P_{\mathrm{BB2}}.\mathtt{open}$	$v_{\mathtt{BB2.open}}$
R34	$P_{\mathrm{BB2}}.\mathtt{close}$	$v_{\mathtt{BB2.close}}$
R35	$P_{\mathrm{BD}}.\mathtt{open}$	$v_{\mathtt{BD.open}}$
R36	$P_{\mathrm{BD}}.\mathtt{close}$	$v_{\mathtt{BD.close}}$
R37	$P_{\mathrm{BD}}.\mathtt{stopOpen}$	$v_{\mathtt{BD.stopOpen}}$

(j) The requirements in the safety part and in the regular part are (Eqs. 18 and 19):

$$\mathcal{R}_{\mathrm{S}} = \{\mathrm{R11, R12, R15, R16, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31} \\ \mathrm{R32, R33, R34, R35, R36, R37}\}$$

$$\mathcal{R}_{\mathrm{R}} = \{\mathrm{R7, R8, R9, R10, R13, R14, R19, R20}\}$$

(k) The variables that have to be communicated are derived (Eqs. 20 and 21). There are no variables that have to be communicated via D_{S}. The variables that are communicated via D_{R} are:

$$D_{\mathrm{R}} = \{v_{\mathtt{SS.on}}, v_{\mathtt{SS.off}}, v_{\mathtt{BB1.open}}, v_{\mathtt{BB1.close}} \\ v_{\mathtt{BB2.open}}, v_{\mathtt{BB2.close}}, v_{\mathtt{BD.open}}, v_{\mathtt{BD.close}}, v_{\mathtt{BD.stopOpen}} \\ P_{\mathrm{GUI}}.\mathtt{StopBarriers}, P_{\mathrm{GUI}}.\mathtt{StopBridgeDeck}\}$$

Appendix A. List of symbols

Model symbols	
$\mathcal{P}$	Plant model
$\mathcal{R}$	Requirements model
Σ	Events
c	Condition
i	Boolean input variable
L	Locations
P	Component model
R	Event-condition requirement
T_Q	Hardware mapping
v_l	Location reference
Split symbols	
Σ_{R}	Set of events in the regular part
Σ_{S}	Set of events in the safety part
D_{R}	Set of regular data buffer variables
D_{S}	Set of safety data buffer variables
I_{R}	Set of regular input image variables
I_{S}	Set of safety input image variables
Q_{R}	Set of regular output image variables
Q_{S}	Set of safety output image variables
$\mathcal{R}_D$	Set of requirements via communication
$\mathcal{R}_{\mathrm{R}}$	Set of requirements in the regular part
$\mathcal{R}_{\mathrm{S}}$	Set of requirements in the safety part
$\mathcal{R}'_{\mathrm{R}}$	Set of regular requirements
$\mathcal{R}^{\mathrm{R}}_{\mathrm{R}}$	Set of regular requirements for Σ_{R}
$\mathcal{R}^{\mathrm{S}}_{\mathrm{R}}$	Set of regular requirements for Σ_{S}
$\mathcal{R}^{\mathrm{SR}}_{\mathrm{R}}$	Set of regular requirements for Σ_{S} in the regular part
$\mathcal{R}^{\mathrm{SS}}_{\mathrm{R}}$	Set of regular requirements for Σ_{S} in the safety part
$\mathcal{R}'_{\mathrm{S}}$	Set of safety requirements
S	Supervisor
c_σ	Regular condition for event σ
v_σ	Evaluation result of c_σ

Appendix B. SPLC implementation code

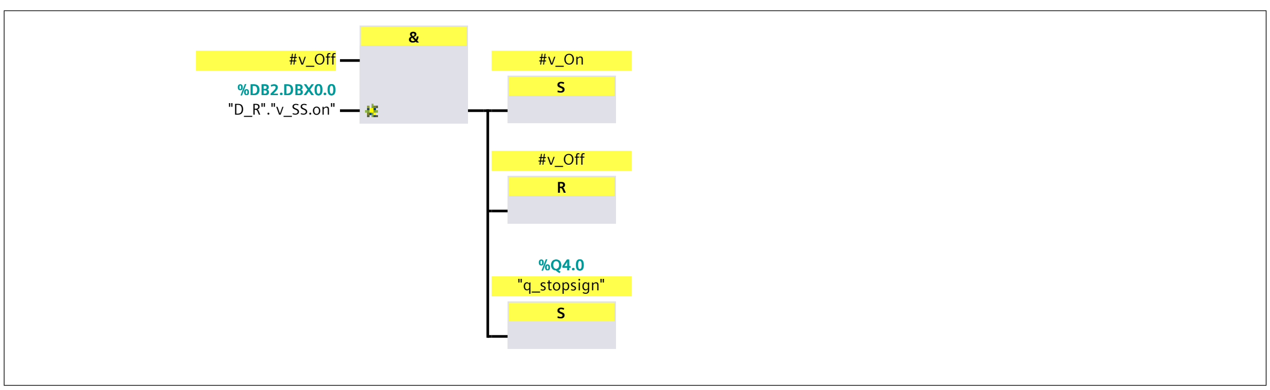
In this section, the function block diagrams (FBDs) implemented in the safety part of the PLC are given. First, the stopsign FBDs are given, then the barrier FBDs, and lastly the bridge FBDs. Here, the FBDs are implemented in TIA Portal Version 15 from Siemens.

P_SS [FB2]

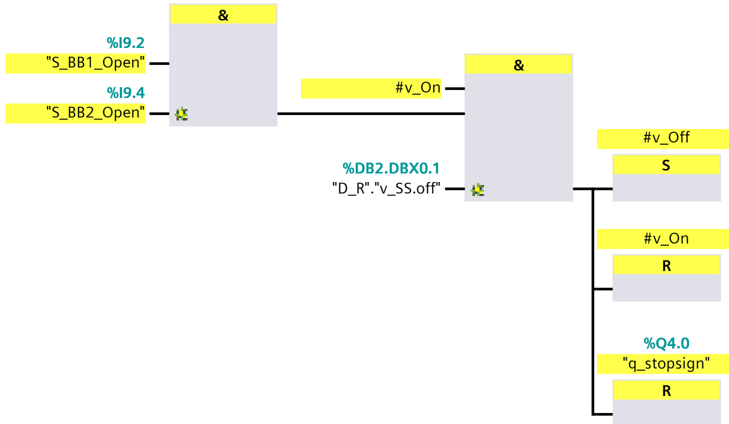
P_SS Properties					
General					
Name	P_SS	**Number**	2	**Type**	FB
Language	FBD	**Numbering**	Automatic		
Information					
Title		**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Offset	Default value	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
Input									
Output									
InOut									
▼ Static									
v_Off	Bool	0.0	true	True	True	True	False		
v_On	Bool	0.1	false	True	True	True	False		
Temp									
Constant									

Network 1: SS.on



Network 2: SS.off

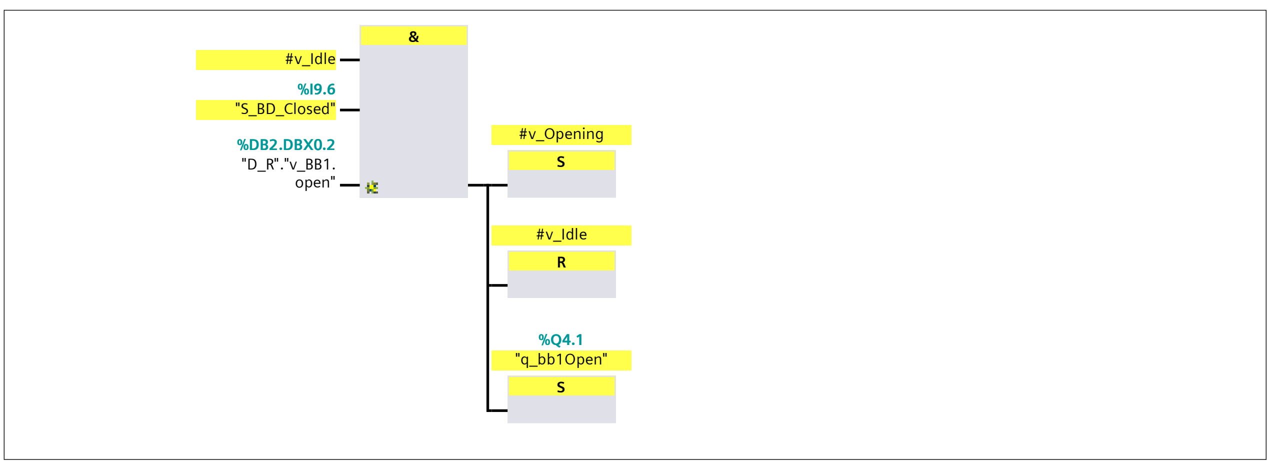

&
%I9.2
"S_BB1_Open"
%I9.4
"S_BB2_Open"
&
#v_On
%DB2.DBX0.1
"D_R"."v_SS.off"
#v_Off
S
#v_On
R
%Q4.0
"q_stopsign"
R

P_BB1 [FB3]

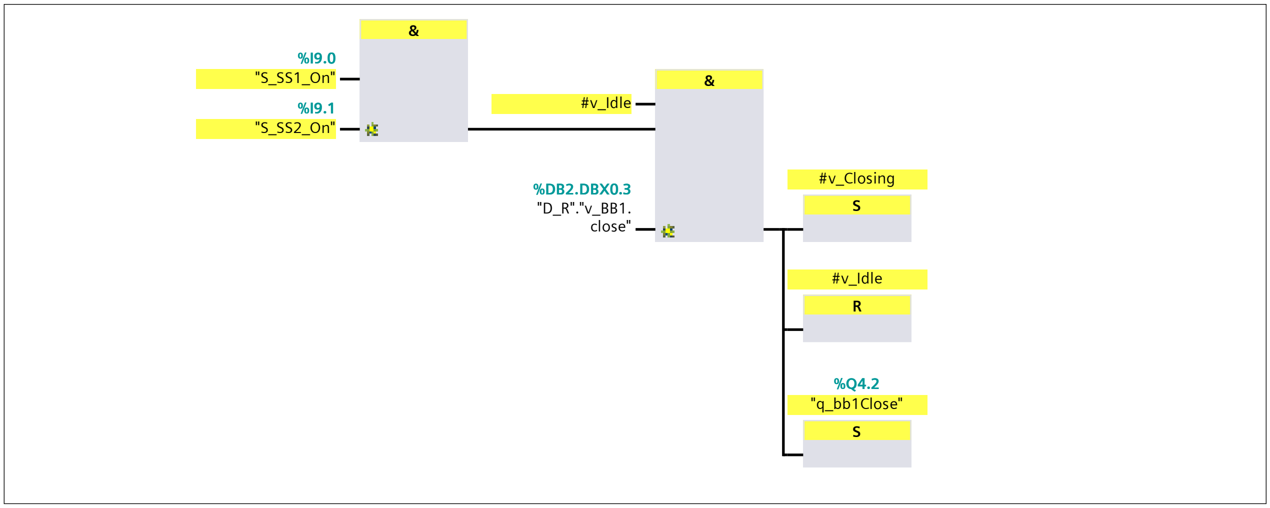
P_BB1 Properties					
General					
Name	P_BB1	**Number**	3	**Type**	FB
Language	FBD	**Numbering**	Automatic		
Information					
Title	boombarrier1	**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Offset	Default value	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
Input									
Output									
InOut									
Static									
v_Opening	Bool	0.0	false	True	True	True	False		
v_Idle	Bool	0.1	true	True	True	True	False		
v_Closing	Bool	0.2	false	True	True	True	False		
Temp									
Constant									

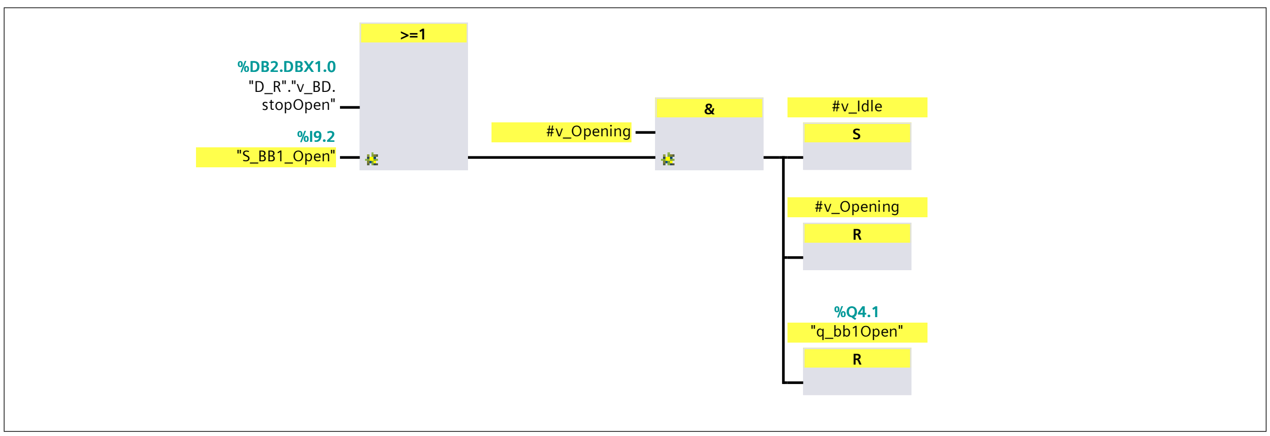
Network 1: BB1.open



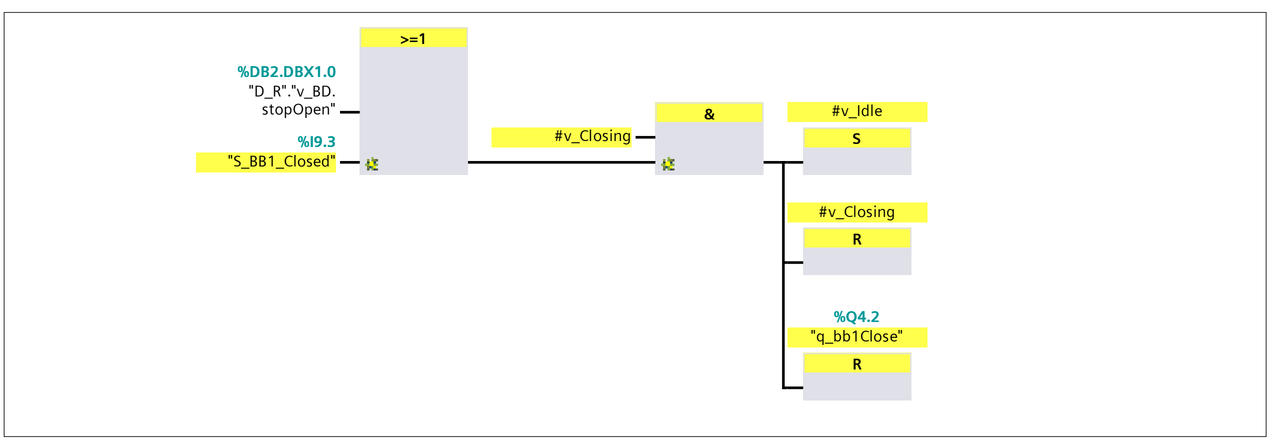
Network 2: BB1.close



Network 3: BB1.stopOpen



Network 4: BB1.stopClose



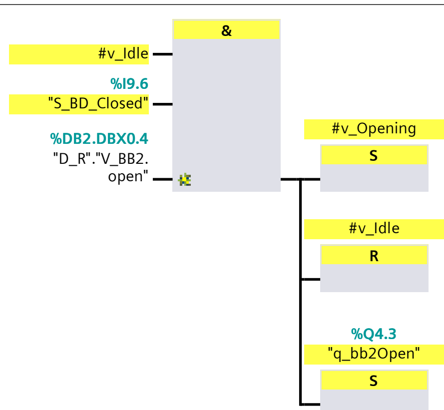
WODES / PLC_1 [CPU 315F-2 PN/DP] / Program blocks

P_BB2 [FB4]

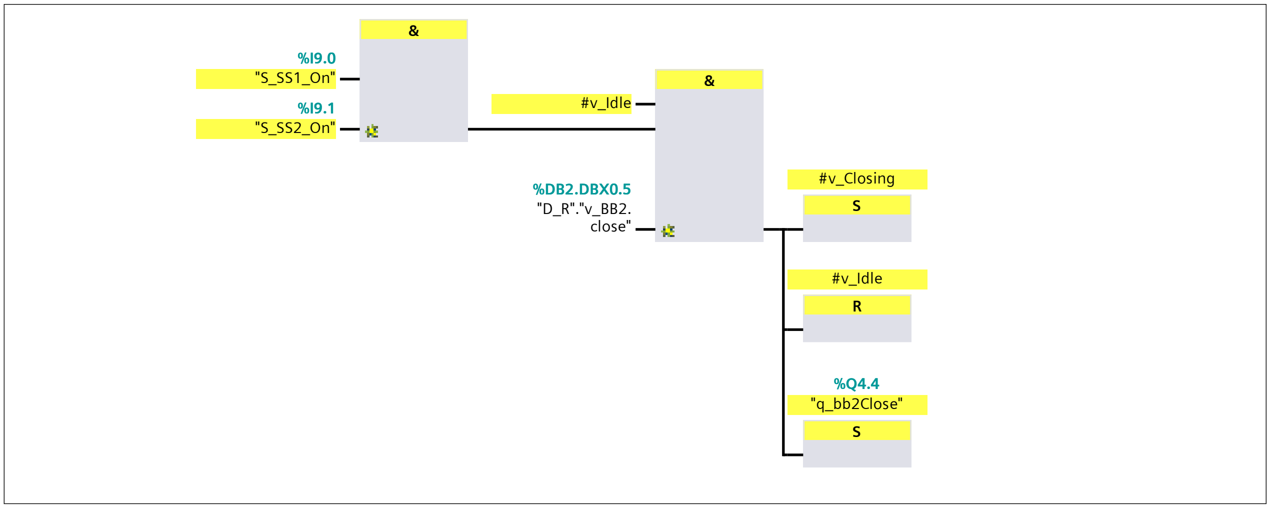
P_BB2 Properties					
General					
Name	P_BB2	**Number**	4	**Type**	FB
Language	FBD	**Numbering**	Automatic		
Information					
Title		**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

P_BB2									
Name	**Data type**	**Offset**	**Default value**	**Accessible from HMI/OPC UA**	**Writable from HMI/OPC UA**	**Visible in HMI engineering**	**Setpoint**	**Supervision**	**Comment**
Input									
Output									
InOut									
▼ Static									
v_Opening	Bool	0.0	false	True	True	True	False		
v_Idle	Bool	0.1	true	True	True	True	False		
v_Closing	Bool	0.2	false	True	True	True	False		
Temp									
Constant									

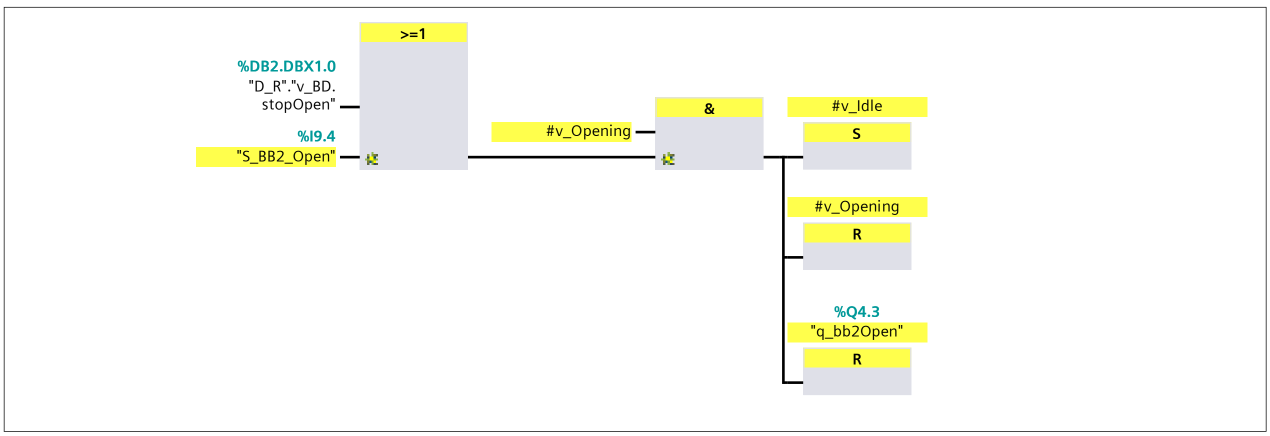
Network 1: BB2.open



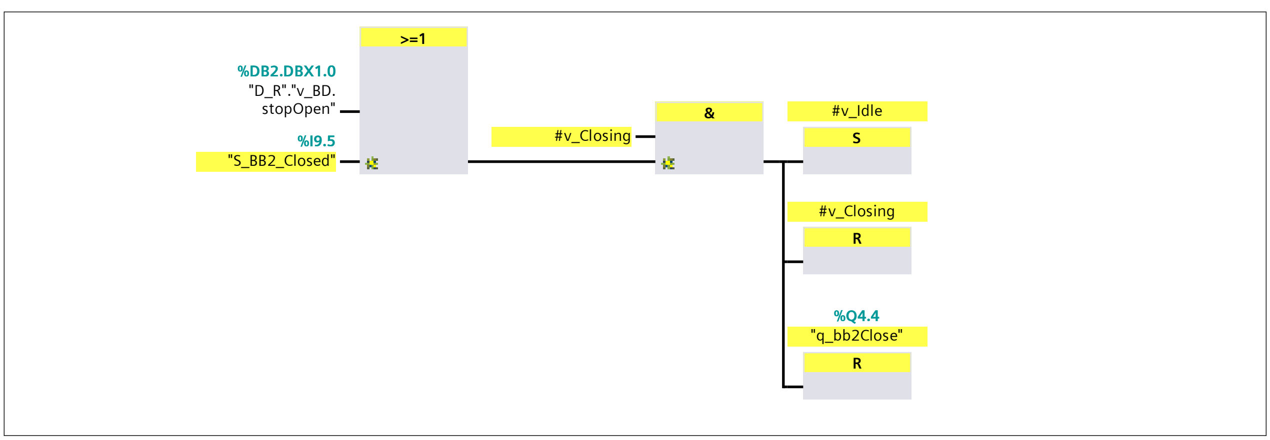
Network 2: BB2.close



Network 3: BB2.stopOpen



Network 4: BB2.stopClose

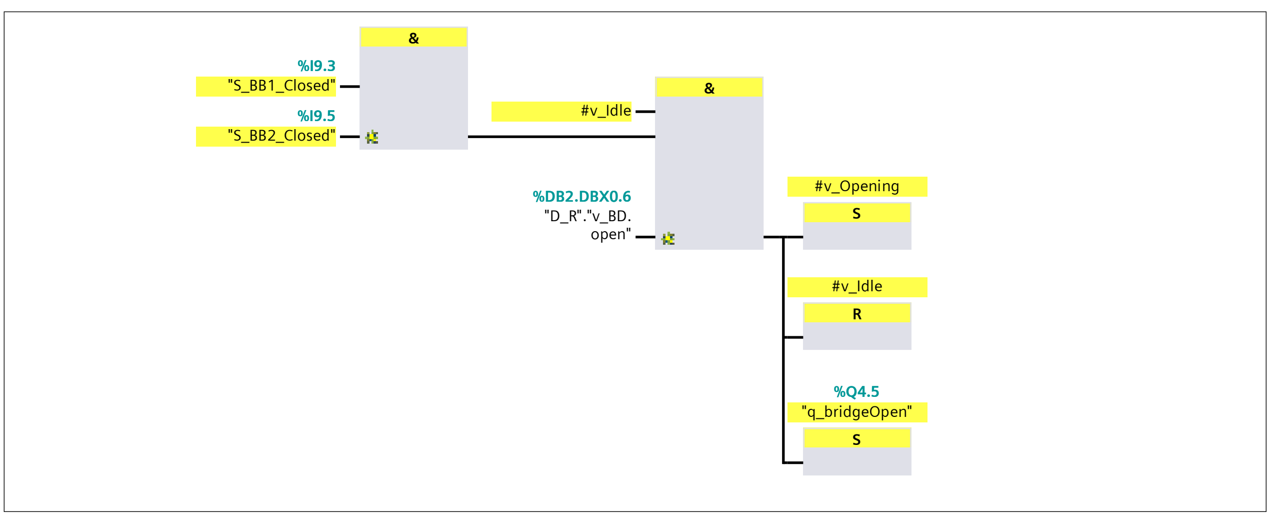


P_BD [FB5]

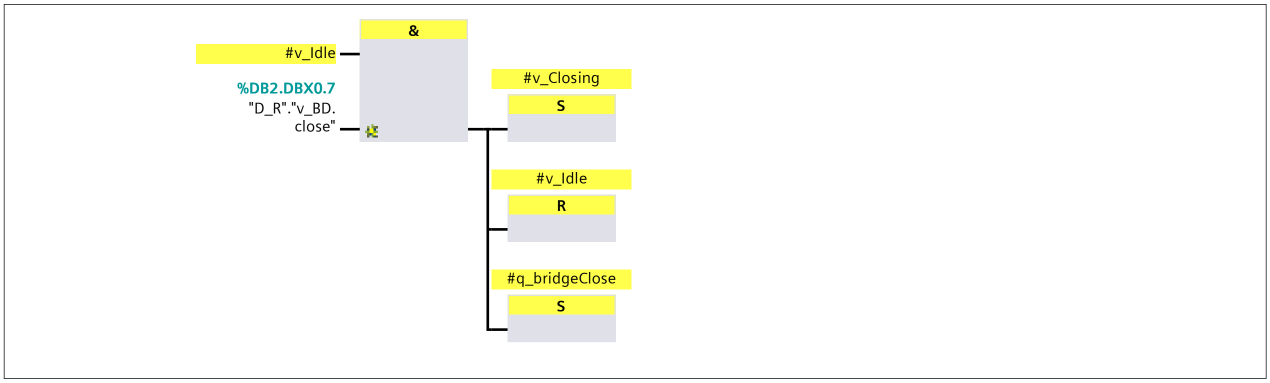
P_BD Properties					
General					
Name	P_BD	**Number**	5	**Type**	FB
Language	FBD	**Numbering**	Automatic		
Information					
Title		**Author**		**Comment**	
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Offset	Default value	Accessible from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engineering	Setpoint	Supervision	Comment
Input									
Output									
InOut									
▼ Static									
v_Opening	Bool	0.0	false	True	True	True	False		
v_Idle	Bool	0.1	true	True	True	True	False		
v_Closing	Bool	0.2	false	True	True	True	False		
q_bridgeClose	Bool	0.3	false	True	True	True	False		
Temp									
Constant									

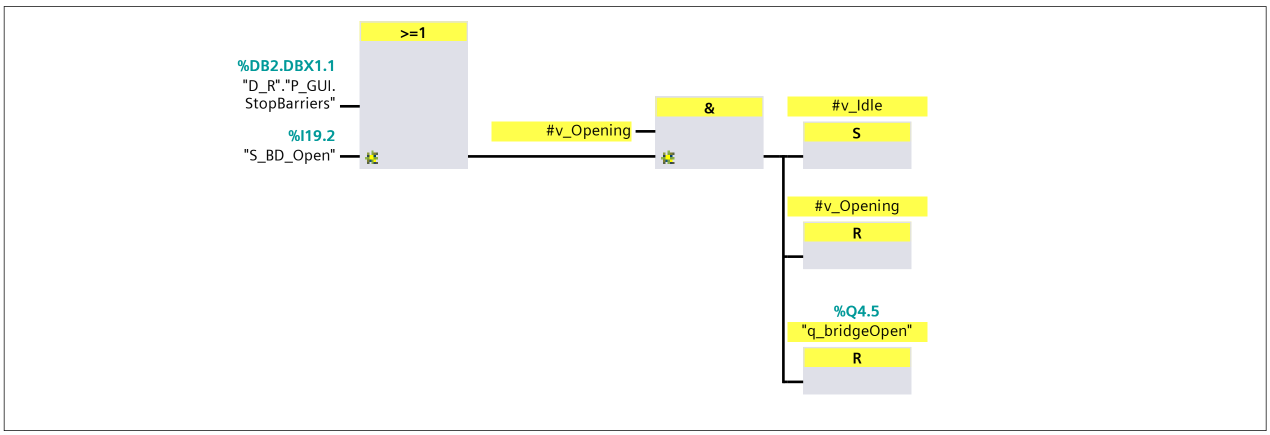
Network 1: BD.open



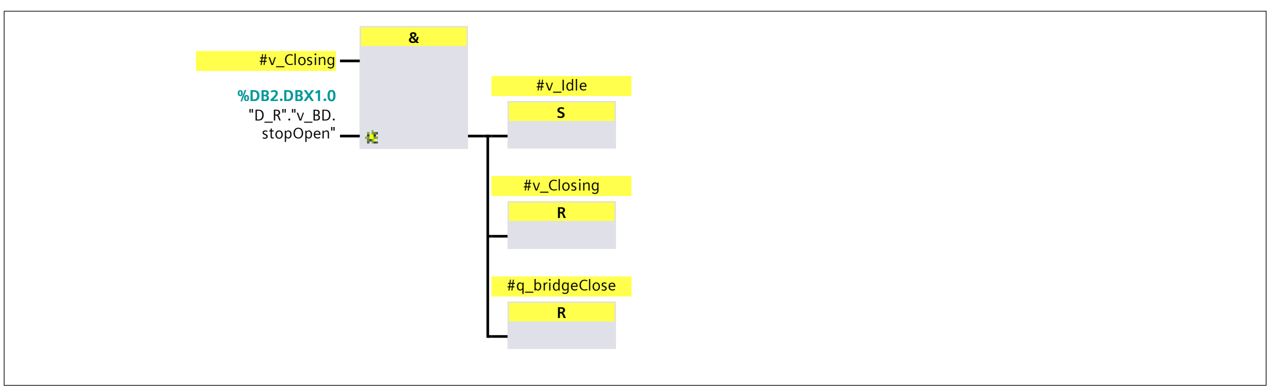
Network 2: BD.close



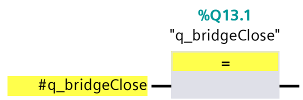
Network 3: BD.stopOpen



Network 4: BD.stopClose



Network 5: BD.stopClose - 2


%Q13.1
"q_bridgeClose"
=
#q_bridgeClose